# الفصل الأول ..... تمثيل البيانات

## كيف تجرى عملية حسابية في ال Visual Basic.net

- في البداية تستقبل ما يكتبه المستخدم باستخدام أداة الكتابة Text Box وهذا يمثل المدخلات (مدخلات حرفية) تحول إلي رقمية .
- ثم يتم اجراء العمليات الحسابية عليها في ذاكرة الحاسب .
- واخيراً تقوم بعرض النتائج بالتنسيق المناسب .

## البيانات الرقمية :

- هى البيانات التى يتم إجراء العمليات الحسابية عليها ، ومن أمثلتها " معدل الفائدة " الذي يعطيه البنك هو بيان رقمى لأنه يتم إجراء عمليات حسابية عليه .

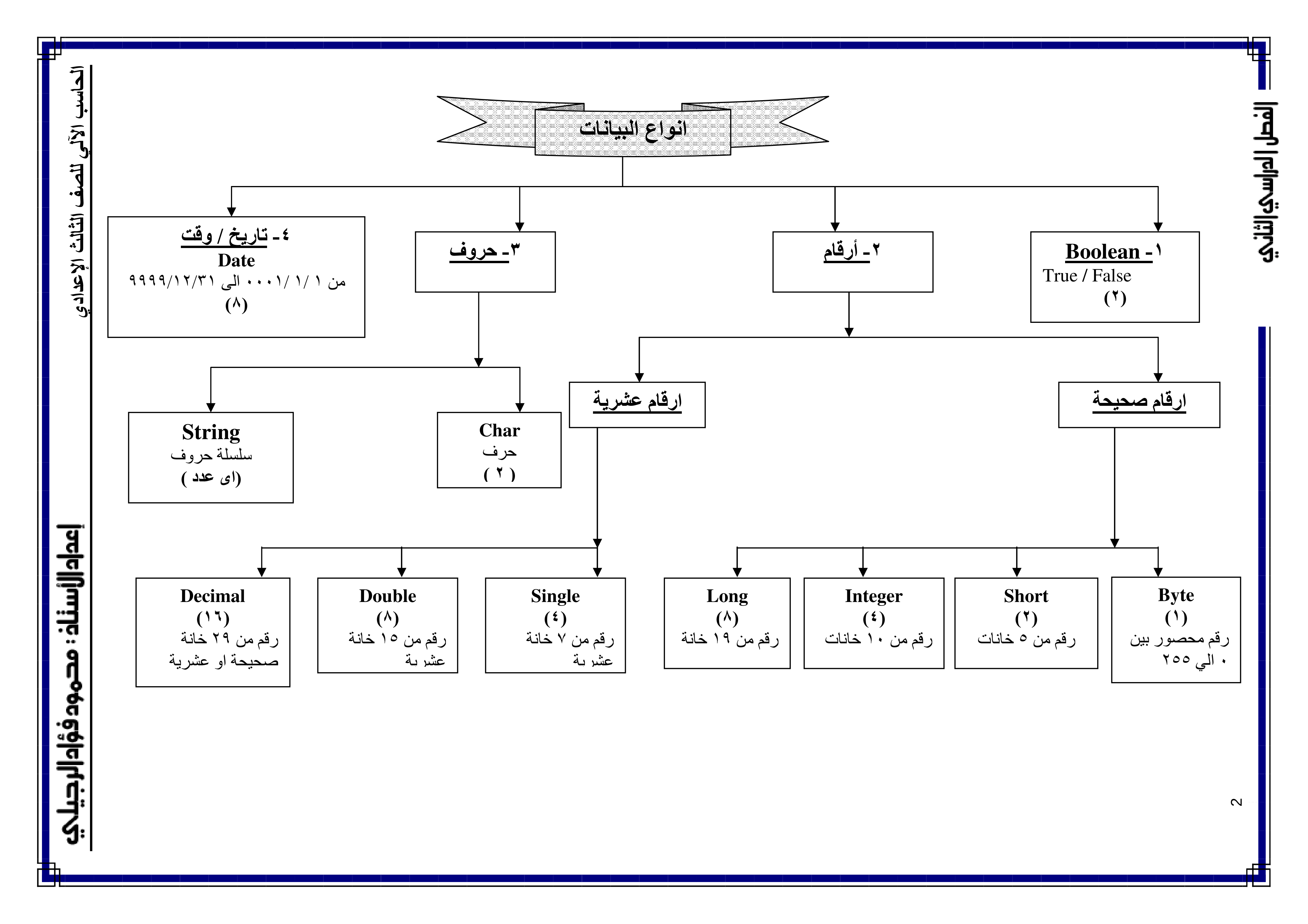
## البيانات الحرفية :

- هى البيانات التى لا يجوز إجراء العمليات الحسابية عليها حتى لو كانت هذه البيانات أرقام ، ومن أمثلتها " رقم الهاتف " هو بيان حرفى لا يجوز إجراء عمليات حسابية عليه على الرغم من أنه يتكون من أرقام

## ❖ البيانات الثابتة و المتغيرة

- اى بيانات تخزن في ذاكرة الحاسب الداخلية .
- هذه ا لذاكرة تتكون من ملايين الخلايا المتساوية و تسمى كل خلية (Byte ) .
- يمثل عدد الخلايا حجم الذاكرة .
- تسمح كل لغات البرمجة بإعطاء اسماء للخلايا التى سيتم تخزين البيانات بها حتى يسهل تتبعها و التعامل معها .
- يعطى اسم واحد لكل خلية أو مجموعة من الخلايا ما دامت تحمل قيمة واحدة . ولذلك

١. اذا كانت القيم المخزنة تتغير اثناء تشغيل البرنامج فإن اسماء الخلايا التى تشغله هذه القيم يطلق عليها متغيرات Variables .
٢. اذا كانت القيم لا تتغير اثناء تشغيلها البرنامج فإن اسماء الخلايا التى تشغلها هذه القيم يطلق عليها الثوابت Constants ، وتعتبر قيمة " ط " والتى تساوى ٣.١٤ مثال للقيم الثابته .
٣. أما عملية إعطاء أسماء للخلايا فيطلق عليها " إعلان المتغيرات "
٤. وإذا كانت الخلايا ستحمل قيما ثابته يطلق عليها إعلان الثوابت .


انواع البيانات
١- Boolean
True / False
(٢)
٢- أرقام
٣- حروف
٤- تاريخ / وقت
Date
من ١ /١ /٠٠٠١ الى ٩٩٩٩/١٢/٣١
(٨)
ارقام صحيحة
ارقام عشرية
Char
حرف
( ٢ )
String
سلسلة حروف
(اى عدد )
Byte
(١)
رقم محصور بين
٠ الي ٢٥٥
Short
(٢)
رقم من ٥ خانات
Integer
(٤)
رقم من ١٠ خانات
Long
(٨)
رقم من ١٩ خانة
Single
(٤)
رقم من ٧ خانة
عشر ية
Double
(٨)
رقم من ١٥ خانة
عشر ية
Decimal
(١٦)
رقم من ٢٩ خانة
صحيحة او عشرية

## ❖ قواعد تكوين الأسماء في ال VB.net:-

### ▪ أولا: قواعد يجب الالتزام بها :-

١- يجب ان يبدأ الاسم بحرف من حروف الانجليزية ( A- Z ) أو (a-z ) ولا يجوز أن يبدأ برقم أو أى رمز آخر ولا يجوز إستخدام اللغة العربية . .

٢- يأتي بعد الحرف الأول أي عدد من الحروف او الأرقام .

٣- لا يسمح ان يحتوى على رموز او علامات او مسافات ماعدا علامة ال _ Underscore .

٤- ممنوع استخدام الكلمات المحجوزة فى VB.netمثل **Data – Short – Double – Decimal – Integer – Long – Case – Single – Boolean – Byte – Char Const – Mod – MsgBox – String – Dim – Print– Color – Me VbCrLf Then – For – Next – To – Step** .

٥- **عدم استخدام إسم واحد مرتين .**

**ماذا يحدث لو لم نلتزم بهذه القواعد ؟**

فإن البرنامج لن يعمل وسيشير المترجم/ المفسر إلى خطأ فى القواعد ( syntax error) .

### ▪ ثانيا : قواعد يستحسن الالتزام بها :-

١- اختر أسماء ذات معنى مناسب مثل الإسم Sum للمجموع .

٢- فى الكلمات التى تتكون من مقطعين يمكن ان يبدأ كل مقطع بحرف Capital .

٣- يفضل ان يبدأ الاسم بـــ ٣ احرف تعطى انطباع عن نوع المتغير ( الإختصارات ) .

| أمثلة لأسماء صحيحة | أمثلة لأسماء غير صحيحة |
|---|---|
| FirstName | 2ndWorldWar ………………… |
| Ahmed2000 | Spen.Mon ……………….. |
| PopulationOfEgypt | Birth Day ……………….. |
| Spent_mony | Double ………………… |

## ❖ الثوابت :-

**التعريف : هى طريقه لإعطاء أسماء للقيم التى لاتتغير أثناء تنفيذ البرنامج . ( عكس المتغيرات )**

**أنواع الثوابت :**

- **ثوابت جواهرية:-**

هي الثوابت الموجودة فى بناء و جوهر نظام ال VB.net
أي يستطيع البرنامج التعامل معها بدون اعلان

**مثال** ثوابت الألوان Color.Black Color.Red Color.Blue

- **ثوابت مسماه :-**

هي الثوابت التي يقوم المبرمج بتعريفها طبقا لاحتياجات البرنامج و هذه تحتاج إلي إعلان عنها قبل استخدامها و إلا حدث خطأ

**الإعلان عن الثوابت**

```
Const Name as String = " Ahmed"
Const Mass AS Decimal = 123.5 d
Const Mass as Single = 0.123 f
```

**لاحظ أن :-**

١. لابد من استخدام كلمة Const للإعلان عن الثابت و هي من الكلمات المحجوزة .
٢. لابد أن تكتب ( = ) ثم بعدها قيمة هذا الثابت .
٣. هذه القيمة تظل ثابتة لا يمكن تغيرها إثناء تشغيل البرنامج .

**مميزات استخدام الثوابت**

١- تجعل البرنامج أكثر فهما عند قراءته .
٢- تقلل الخطأ عند الرغبه فى تغيير أى قيمه .

**المتغيرات**

**تعريف المتغير :**

**مكان داخل الذاكره له إسم ويخزن بداخله قيمة قد تتغير فى أى وقت .**

**الإعلان عن المتغيرات**

```
Dim school as Integer
Dim Name as String
Dim Address as String = "Cairo"
Dim Price as Decimal = 123
```

**لاحظ ان :-**

١- لابد من استخدام كلمة Dim للإعلان عن المتغير .

٢- يمكن أن تكتب ( = ) و بعدها قيمة و هذه القيمة ستكون قيمة البداية .

أو يمكن ان تهملها و لا تستخدم (=) و فى هذه الحالة

* المتغيرات الرقمية تبدأ بقيمة صفر

* المتغيرات الحرفية ستكون فارغة

**الإعلان عن أكثر من متغير باستخدام أمر Dim واحد**

```
Dim Child as Integer = 123, Name as String = "Ahmed"
Dim Child , Name , Address as Integer
```

❖ **أمر التخصيص ( الإحلال ) :-**

إن أهم دور للمتغيرات هو تخزين البيانات بها ولكي نخزن قيمة فى متغير نستخدم أمر التخصيص

**قيمة = اسم المتغير**

علامة = تسمى معامل التخصيص

( اى يتم حجز مكان فى الذاكرة يسمى باسم المتغير و توضع داخله القيمة )

**يجب مراعاة الاتى:-**

١. اسم المتغير الذى نريد التخزين فيه يكون شمال (=) .

٢. القيمة المراد تخزينها تكون يمين (=) وتكون من نوع ملائم لنوع المتغير .

٣. القيمة يمكن ان تكون :-

a. قيمة مجردة مثل X=5

b. متغير من نوع أخر X=Y

c. عملية حسابية X=2Y+Z

**مثال**

كميه

**Price = Quantity*UnitPrice**

حيث أن:

**Quantity = 100**
**UnitPrice = 5.5**
**Price = 0.0**

**ولتنفيذ هذا الأمر سيقوم الحاسب بعمل الآتي:-**

أ- استبدال القيمتين 100 & 5.5 بالمتغيرين UnitPrice & Quantity بالترتيب فيصبح الطرف الأيمن للعلامة "=" هو 5.5 * 100

ب- تنفيذ عملية الضرب وحساب النتيجة، فتصبح القيمة تساوى (550)

ت- تخزين النتيجة ( 550 ) فى Price

ونلاحظ في هذا الأمر أن القيمة UnitPrice & Quantity لم تتغير وذلك لاننا فقط نستعمل قيمة المتغيرات التي علي يمين "=" واستعمال القيم لا يؤثر فيها ولا يغيرها بكل تأكيد؛ أما قيمة Price فإنها تغيرت نتيجة تخزين النتيجة بها، وذلك يعني أن النتيجة ( 550) قد حلت محل القيمة القديمة للمتغير Price و التى كانت صفرا .

- **استعمال أمر التخصيص مع المتغيرات الحرفية**

```
FirstName ="Hany"
SecondName = "Ahmed"
Fullname = "Welcome"&FirstName&"  "&SecondName
```

تكون النتيجة

Welcome Hany Ahmed

- قمنا بعملية ربط بين المتغيرات باستخدام العلامة &
- "   " تعنى انه سيترك مسافة فارغة مقدارها مسافة واحدة او عدة مسافات
- "" تعنى قيمة حرفية فارغة ( طولها صفر ) .

## ❖ التحويل بين الأنواع المختلفة للبيانات باستخدام Parse Method

**الشكل العام لل Parse Method**

١. لتحويل قيمة حرفية الي عدد صحيح Integer

Integer.Parse (القيمة الحرفية )

٢. لتحويل قيمة حرفية الى عدد عشرى Decimal

Decimal.Parse (القيمة الحرفية )

## ❖ تحويل البيانات الرقمية إلي بيانات حرفية

باستعمال To string ( )

يتم التحويل من نوع إلى أخر بإستخدام To ويسمى هذا التحويل بالتحويل التلقائى

## ❖ العمليات الحسابية :-

| المعامل المؤثر | العملية |
|---|---|
| + | جمع |
| − | طرح |
| * | ضرب |
| / | قسمة عادية |
| \ | قسمة أعداد صحيحة |
| mod | باقي القسمة |
| ^ | أسس |

**قسمة الأعداد الصحيحة وباقي القسمة**

إذا قسمنا العدد الصحيح ( Integer) ٤٨ على العدد الصحيح ٥ فتكون النتيجة كالآتى :-

١. قسمة عادية

$48 / 5 = 9 \frac{3}{5}$

٢. قسمة اعداد صحيحة

( اى اهمال باقى القسمة)

48 \ 5 = 9

٣. اذا اردنا الباقى فيمكن الحصول عليه كالآتى :-

48 mod 5 = 3

### ❖ ترتيب تنفيذ العمليات الحسابية :-

ترتيب تنفيذ العمليات الحسابية داخل VB يتم كالآتى : حتى يتم الحصول على النتائج الصحيحة

1. **العمليات التى بين الاقواس ( الداخلية اولا ثم الخارجية )**
2. **عمليات الاسس ^**
3. **عمليات الضرب و القسمة * / ( الاولوية من اليسار ثم اليمين )**
4. **قسمه الأعداد الصحيحه \ .**
5. **باقى القسمه Mod .**
6. **عمليات الجمع و الطرح + - ( الاولوية من اليسار الي ليمين )**

مثال :-

ما هي نتيجة تنفيذ العمليات الآتية بافتراض القيم التالية للمتغيرات :-

X = 2     Y = 3     Z = 4

A) X * ( Y + 1 )

B) Y^ X * X + Z *3

الحل

(A

١- نستبدل كل متغير بقيمته     2* ( 3 + 1)

٢- حساب ما بداخل الاقواس اولا     2 * ( 4 )

٣- تنفيذ عملية الضرب فيكون الناتج     ( 8 )

( B

1. نستبدل كل متغير بقيمته     3^ 2 * 2 + 4 * 3
2. يتم حساب الاس اولا     9 * 2 + 4 * 3
3. تنفيذ عملية الضرب التى في اليسار     18 + 4 * 3
4. تنفيذ عملية الضرب التى في اليمين     18 + 12
5. تنفيذ الجمع ليكون الناتج     30

## ❖ مجموعة أوامر Try / Catch :-

_ تستخدم لاكتشاف الأخطاء التي تحدث أثناء تشغيل البرنامج و التعامل معها .

- **الفكرة الأساسية لها :-**

| | |
|---|---|
| Try | مجموعة من الأوامر التى لابد ان يبذلها المستخدم بدون أخطاء |
| Catch | كود يكتب و لا ينفذ الا ي حالة حدوث اخطاء في الاوامر السابقة و غالبا يكون رسالة |
| End Try | بقية أوامر البرنامج |

- اذ قمنا بتنفيذ الاوامر بعد Try بدون اخطاء ينتقل البرنامج الي الاوامر بعد End Try مباشرة
- اذا تم تنفيذ الاوامر بعد Try بأخطاء ينتقل البرنامج الي الاوامر بعد Catch ليعرض الرسالة الموجودة بها و لا يتابع تنفيذ الاوامر بعد ذلك تصحيح الخطأ .

# أسئلة الكتاب للفصل الأول

## السؤال الأول : اختر الإجابة الصحيحة من الاختيارات المتاحة

1- العملية التي يتم بها التحويل من نوع بيانات ضيق إلى نوع أكثر اتساعا يطلق عليها اسم

أ) تحويل البيانات Data Transformation

ب) التحويل التلقائي Implicit Conversion

ج) تغيير البيانات Data Change

د) التحويل الغير مباشر Explicit Conversion

2- الرمز " " في لغة VB.Net يرمز به إلى :

أ) القيمة صفر  ب) السلسلة الحرفية الفارغة  ج) الرمز الخالي  د) لا شيء مما سبق

3- التعبير الذي يستخدم في الإعلان عن ثلاثة متغيرات من النوع رقم صحيح Integer ومتغيرين من النوع سلسلة حرفية String هو :

أ) Dim N1,N2,N3 As Integer , S1,S2 As String

ب) Dim N1,N2,N3 As Integer
Dim S1,S2 As String

ج) Dim N1 As Integer
Dim N2 As Integer
Dim N3 As Integer
Dim S1 As String
Dim S2 As String

د) كل من أ و ج

4- عند وضع قيمة في عنوان في ذاكرة الحاسب فإن هذه القيمة :

أ) تضاف إلى القيمة السابقة الموجودة في هذا العنوان .

ب) تحل محل القيمة السابقة الموجودة في هذا العنوان

ج) يتم قسمتها على القيمة السابقة الموجودة في هذا العنوان

د) يتم طرحها من القيمة السابقة الموجودة في هذا العنوان .

5- عند قراءة قيمة من عنوان من الذاكرة يحدث الآتي :

أ) يتم استبدال الموجود في العنوان بالقيمة صفر

ب) لا تتأثر محتويات عنوان الذاكرة

ج) يتم نقل القيمة إلى عنوان آخر

د) يتم استبدال الموجود في العنوان بسلسلة حرفية فارغة Empty String

6- يتم احتساب قيمة التعبير الموجود على يمين علامة "=" في التوقيت الآتي :

أ) بعد تنفيذ التخصيص

ب) قبل تنفيذ التخصيص

ج) في نفس توقيت التخصيص

د) في توقيت آخر غير التوقيتات المذكورة

7- القسمة الرقمية Integer Division يعبر عنها عن طريق :

أ) Mod ب) Div ج) \ د) ÷

8- يبدأ الإعلان عن المتغيرات باستخدام الكلمة الآتية :

أ) Dim ب) Declare ج) As د) Variable

9- لتحويل قيمة عددية إلى نوع سلسلة حرفية String يتم استخدام الدالة الآتية :

أ) MakeString

ب) ToString

ج) ChangeToString

د) لا يتم استخدام أي من الدوال السابقة

10- يتم احتساب التعبيرات الحسابية Arithmetic Expressions بطريقة :

أ) احتساب من اليمين إلى اليسار

ب) احتساب من اليسار إلى اليمين

ج) من المستوى الأعلى في الأسبقية إلى المستوى الأقل

د) من المستوى الأقل في الأسبقية إلى المستوى الأعلى

## السؤال الثانى : مجموعة الأوامر التالية تؤثر على قيم المتغيرات num1 , num2 , num3 تتبع هذه الأوامر موضحًا قيم كل المتغيرات في كل خطوة :

```
Dim num1,num2,num3 as Integer
Num3 = num1 + num2 + num3
Num2 = num2 + 2
Num1 = 4
Num2 = num1 – 3
Num1 = num1 + num1
Num2 = num2 + num3
Num2 = 5 * num2 / num1
```

## الحل

| الأمر | Num1 | Num2 | Num3 |
|---|---|---|---|
| Dim num1,num2,num3 as Integer | 0 | 0 | 0 |
| Num3 = num1 + num2 + num3 | 0 | 0 | 0 |
| Num2 = num2 + 2 | 0 | 2 | 0 |
| Num1 = 4 | 4 | 2 | 0 |
| Num2 = num1 – 3 | 4 | 1 | 0 |
| Num1 = num1 + num1 | 8 | 1 | 0 |
| Num2 = num2 + num3 | 8 | 1 | 0 |
| Num2 = 5 * num2 / num1 | 8 | 0 | 0 |

## السؤال الثالث : ما الخطأ بالبرنامج التالي

```
Dim Number1 As Integer
Number1 = ( 4 * 3^2 ) / ( 10 Mod 3 – 1 )
```

## الحل

الخطأ بالبرنامج السابق هو أن ناتج العملية الحسابية ستكون كالتالي :

Number1 = ( 4 * 9 ) / ( 1 – 1 ) = 36 / 0

والخطأ هو القسمة على صفر حيث أن ليس لها معنى

**تدريب** :**أوجد ناتج تنفيذ العمليات التالية داخل** VB.Net :

- 8 + 5 * 3
- (8 + 5) * 3
- 16 / 4 + 8 + 6 * 8

## تمارين

**١- أوجد ناتج تنفيذ العمليات التالية فى لغة VB.Net**

- X = 10 + 3 / 3 * 2 ^ 2 mod 4
- Y = 16 ^ 0.5 mod 3
- Z = 5 + 5 – 10 * 3 / 6
- X = ( 3 + 7 ) / 2 + 3 / 6 * 2
- Y = 1 + ( 2 + 2 ) ^ 2 / 4 * 2
- Z = 10 \ 3 * 5 / 3
- X = 9 + 9 / 3 / ( 9 mod 4 -1 )
- Y = 16 + 5 – 10 / ( 2 + 3)
- Z = 15 / 9 ^ 0.5 \ 3
- X = 13 – 3 / 3 + 2 * 5

**٢- إذا علمت أن A = 1 و B = 2 و C = 3 أوجد ناتج كل من العمليات التالية :**

- A ^ 2 – B * C / ( A +2 )
- ( A – 2B) \ ( C + B)
- A + B / B + C
- ( A + 3 * B / C ) ^ 2 – 2 + 3 * B
- 5 * 1 + A / C MOD B
- 10 / 2 * B + C / 2 * B \ A
- A + B - 8 + 9 / 3 MOD 4 ^ 0.5
- A – B / - 1
- - A – C / -2
- - 2 * - A / 2 + 3 * C MOD 2

**٣- ناتج تنفيذ الجملة ( 5*3+2 ) هو 25** (    )

**٤- عملية الأس تلي عملية قسمة الأعداد الصحيحة فى أولويات تنفيذ العمليات الحسابية** (    )

**٥- في أي عملية حسابية إذا وجدت أقواس داخلية وخارجية فان الأقواس الخارجية تنفذ أولاً** (    )

**٦- المعامل " \ " يسبق المعامل " / " فى أولويات تنفيذ العمليات الحسابية** (    )

**٧- القسمة الرقمية Integer Division يعبر عنها عن طريق :**

أ) Mod    ب) Div    ج) \    د) ÷

**٨- ناتج تنفيذ العملية : 5+2*(3+7)/5 هو :**

أ) 14    ب) 5    ج) 12    د) غير ذلك

**٩- العملية التي تسبق عمليتي الضرب والقسمة في أولويات تنفيذ العمليات الحسابية هي العملية :**

أ) الجمع والطرح    ب) حساب باقي القسمة    ج) قسمة الأعداد الصحيحة    د) عمليات الأسس

# الفصل الثاني
# الشروط والقرارات

- **استخدام التعبيرات الشرطية**

**التعبير الشرطي :** هو عبارة عن صياغة للعلاقة بين كميتين

- أي من هاتين الكميتين من الممكن أن تكون متغير أو خاصية أو ثابت أو قيمة مجردة .
- للتعبير الشرطي قيمة تتوقف على صحة العلاقة أو عدم صحتها .
- فإذا كانت العلاقة غير صحيحة فتكون قيمة التعبير الشرطي False .
- والعكس إذا كانت العلاقة صحيحة فتكون قيمة التعبير الشرطي True .
- لصياغة التعبيرات الشرطية فإننا نقوم باستعمال بعض المعاملات تسمى معاملات المقارنة .

**مثال**

**A = 50 B = 60**

| النتيجة | مثال | العلاقة | العلامة |
|---|---|---|---|
| **False** | A > B | اكبر من | > |
| **True** | A < B | اصغر من | < |
| **False** | A = B | يساوي | = |
| **True** | **A < > B** | **لا يساوي** | < > |

- **قاعدة استخدام الأمر If ……. Then ……. Else**

```
If condition ( شرط ) Then
   Statement Block1 ( مجموعة الأوامر رقم ١ )
Else
   Statement Block2 ( مجموعة الأوامر رقم ٢ )
End If
Statement Block3 ( مجموعة الأوامر رقم ٣ )
```

١. إذا كانت قيمة الشرط True سيقوم البرنامج بتنفيذ مجموعة الأوامر رقم ١ ويهمل مجموعة الأوامر رقم ٢ ثم يذهب لتنفيذ مجموعة الأوامر رقم ٣ .
٢. إذا كانت قيمة الشرط False سيقوم البرنامج بتنفيذ مجموعة الأوامر رقم ٢ ويهمل مجموعة الأوامر رقم ١ ثم يذهب لتنفيذ مجموعة الأوامر رقم ٣ .

- **الشكل العام للأمر If ……. Then ……. Else**

```
If condition1 Then
Statement block 1
Else If condition2 Then
Statement block 2
………………………..
………………………..
Else
Statement block n
End If
```

**أي أن التعبيرات الشرطية يتم إيجاد قيمها من أعلى إلى أسفل .**



**مثال**

أكتب برنامجا لحساب الراتب الأسبوعى لموظف بشرط لوكانت ساعات عمل الموظف أقل من أو تساوى 40 ساعة فإن أجرة عن الساعة يكون 5 جنيهات . أما إذا كانت ساعات العمل أكثر من 40 ساعة فإن الراتب يتم حسابه كالآتى :

أ – خمسة جنيهات فى الساعة لأول 40 ساعة .

ب – عشرة جنيهات فى الساعة لما يزيد عن الأربعين ساعة الأولى . أى أن الراتب سيتم حسابه كما يلى :

الراتب = 5×40 + ( ساعات العمل – 40 ) ×10



**الحـــــــــــــــــــل**




```
Dim HoursWorked, OverTime, Salary as Decimal
HoursWorked=Decimal.parse(textbox1.text)
If HoursWorked > 40 Then
        OverTime = 10 * (HoursWorked - 40)
         Salary = 40* 5 + OverTime
Else
          Salary = 5 * HoursWorked
End If
MsgBox("The salary is equal to " & Salary)
```

- **قاعدة استخدام الأمر If ……. Then البسيط**

```
If condition Then
Statement block 1
End If
Statement block 2
```

١. إذا كانت قيمة الشرط True سيقوم البرنامج بتنفيذ مجموعة الأوامر رقم ١ ثم يذهب لتنفيذ مجموعة الأوامر رقم ٢ .
٢. إذا كانت قيمة الشرط False فان لبرنامج سيذهب مباشرة لتنفيذ مجموعة الأوامر رقم ٢ .

**مثال**

```
If Age > 16 Then
MsgBox("You can get an ID card")
End If
```

- **الفرق بين If ……. Then ……. Else و If ……. Then**

١. الأمر If ……. Then ……. Else

سيتم استخدامه في حاله ما إذا كان هناك بديلين ( أ، ب ) وكنا نريد تنفيذ أحد البديلين فاذا كانت قيمة الشرط True يتم تنفيذ الأمر الأول ويتجاهل الثانى وإذا كانت قيمة الشرط False يتم تنفيذ البديل الثانى و هكذا إلى أن نصل إلى End If .

٢. الأمر If ……. Then

يتم استخدامه في حاله ما إذا كان هناك بديل واحد ( أ ) بمعنى انه عند قيمة معينة للشرط يتم تنفيذ البديل (أ) أما عند القيمة الأخرى فلا يتم عمل شئ .

- **الشروط المركبة**

تستخدم في حالة ما إذا كنا سنستخدم قرارا معينا بناءا على تحقق شرطين وليس شرط واحد ويمكننا عمل الشرط لمركب بالربط بين أكثر من شرط بأحد المعاملات التي تسمى المعاملات المنطقية وهي

Or , OrElse , And , AndAlso , XOr , Not

firstInteger = 50
secondInteger = 90

| المعامل المنطقي (Logical operator) | المعنى (Meaning) | مثال (Example) | النتيجه (Result) |
|---|---|---|---|
| **Or** | إذا كان أحد الشرطين أو كلاهما صحيح يكون الشرط صحيح | firstInteger = 50 Or secondInteger = 200 | True Or False ⇒ True |
| **OrElse** | إذا كان الشرط الأول صحيحا فإن النتيجة تكون صحيحه ويكون الشرط الثانى لا قيمه له | firstInteger = 50 Or secondInteger >200 | True OrElse...... ⇒ True |
| **And** | إذا كان كل من الشرطين صحيحا فإن الشرط يكون صحيحا | firstInteger <= 50 And secondInteger < 200 | True And False ⇒ False |
| **AndAlso** | إذا كان الشرط الأول خطأ تكون النتيجة خطأ والشرط الثانى لاقيمة له . | firstInteger <50 And secondInteger = 200 | False AndAlso ...... ⇒ False |
| **Xor** | إذا كان كل من الشرطين صحيحا أو كلاهما خطأ فإن النتيجة تكون خطأ | firstInteger = 50 Or secondInteger = 200 | True Xor True ⇒ False |
| **Not** | تعكس الشرط . إذا كان الشرط صحيحا فإن قيمته خطأ والعكس صحيح . | Not (firstInteger = 50) | Not (True) ⇒ False |

- **مقارنة القيم والمتغيرات الحرفية( String )**

يتم مقارنة المتغيرات الحرفية عن طريق تحديد رتبة كل رمز طبقا لطريقة تخزين الرموز في ذاكرة الحاسب وهذه الطريقة تسمى

ANSI= American National Standards Institute

وهي تعطي كل حرف أو رقم أو رمز خاص رتبة معينة وهذه الرتبة هي رقم ثابت لا يتغير كما في الجدول

| Code | Character | Code | Character | Code | Character | Code | Character |
|---|---|---|---|---|---|---|---|
| 32 | Space | 56 | 8 | 80 | P | 104 | h |
| 33 | ! | 57 | 9 | 81 | Q | 105 | i |
| 34 | “ | 58 | : | 82 | R | 106 | j |
| 35 | # | 59 | ; | 83 | S | 107 | k |
| 36 | $ | 60 | < | 84 | T | 108 | l |
| 37 | % | 61 | = | 85 | U | 109 | m |
| 38 | & | 62 | > | 86 | V | 110 | n |
| 39 | ‘ | 63 | ? | 87 | W | 111 | o |
| 40 | ( | 64 | @ | 88 | X | 112 | p |
| 41 | ) | 65 | A | 89 | Y | 113 | q |
| 42 | * | 66 | B | 90 | Z | 114 | r |
| 43 | + | 67 | C | 91 | [ | 115 | s |
| 44 | , | 68 | D | 92 | \ | 116 | t |
| 45 | - | 69 | E | 93 | ] | 117 | u |
| 46 | . | 70 | F | 94 | ^ | 118 | v |
| 47 | / | 71 | G | 95 | _ | 119 | w |
| 48 | 0 | 72 | H | 96 | ‘ | 120 | x |
| 49 | 1 | 73 | I | 97 | a | 121 | y |
| 50 | 2 | 74 | J | 98 | b | 122 | z |
| 51 | 3 | 75 | K | 99 | c | 123 | { |
| 52 | 4 | 76 | L | 100 | d | 124 | \| |
| 53 | 5 | 77 | M | 101 | e | 125 | } |
| 54 | 6 | 78 | N | 102 | f | 126 | ~ |
| 55 | 7 | 79 | O | 103 | g | 127 | Del |

## الكود آنسى ANSI Code

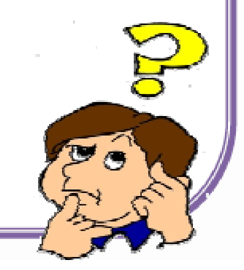
**مثال 2-3 :**

إذا علمنا أن

txtPerson1.Text = "NEHAD" , txtPerson2.Text = "NEHAL"
lblStreet1.Text = "SALAH SALEM" , lblStreet2.Text = "SALAH"
lblCar1.Text = "843335 Cairo" , lblCar2.Text = "Nasr"
txtName1.Text = "ahmed" , txtName2.Text = "Ahmed"

حدد أي من الشروط الآتية ستكون قيمته True وأيها سيكون قيمته False :

i- txtPerson1.Text < txtPerson2.Text
ii- lblStreet1.Text < lblStreet2.Text
iii- lblCar1.Text < lblCar2.Text
iv- txtName1.Text < txtName2.Text



**الحـــــــــل**

أ- الشرط txtPerson2.Text < txtPerson1.Text سيكون "True" وذلك لان " D " في " NEHAD " أقل في الرتبة من "L" في " NEHAL " .

ب- الشرط lblStreet2.Text < lblStreet1.Text قيمته "False" وذلك لأن إحدى القيمتين أقصر من الأخرى وفي هذه الحالة تتم المقارنة كما لو كانت القيمة الأقصر متبوعة بمسافات إلي جهة اليمين؛ وهذا يعنى أن أول مسافة سيتم مقارنتها بحرف أو رمز من القيمة الأطول و لكي نوضح أكثر فإن عملية المقارنه تتم كما لو أننا كنا نقارن

" SALAH SALEM "

مع

" SALAH "

وهذا يعني أن " SALAH SALEM " أكبر من " SALAH" وذلك لأن الحرف "S" في "SALEM" له رتبة أعلي من المسافة المناظرة له.

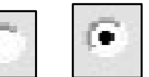
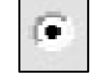
ج - الشرط lblCar2.Text < lblCar1.Text قيمته "True"
وذلك لأن الرقم "8" تمت مقارنته بالحرف "N" وحيث أن كل الأرقام لها رتب أقل من الحروف فان رتبه "8" تكون أقل من رتبة "N" ومن الجدير بالملاحظة هنا أن "Nasr" أكبر من " 843335 Cairo" بالرغم من كونها أقصر لأن العبرة ليست بالطول ولكن برتبه الأحرف أو الرموز المتناظرة .
د- الشرط txtName1.Text < txtName2.Text2 قيمته "False" وذلك لأن رتبه "a" في "ahmed" أعلي من رتبة "A" في "Ahmed".

**<u>ونلاحظ من إجابة السؤال السابق أنه :</u>**

١- عند المقارنة نتجاهل الحروف المتشابهة ونقارن فقط بين الحروف المختلفة تبعا لكل كود مناظر للحرف المختلف .
٢- المسافات أقصر فى المقارنة .
٣- إذا تمت المقارنة بين قيمتين أحدهما يبدأ برقم فتكون هى الاقصر ( التي تبدأ برقم ).
٤- الحروف الصغيره Small تكون أطول من الكبيره Capital .

- **<u>الأداة Radio Button</u>**

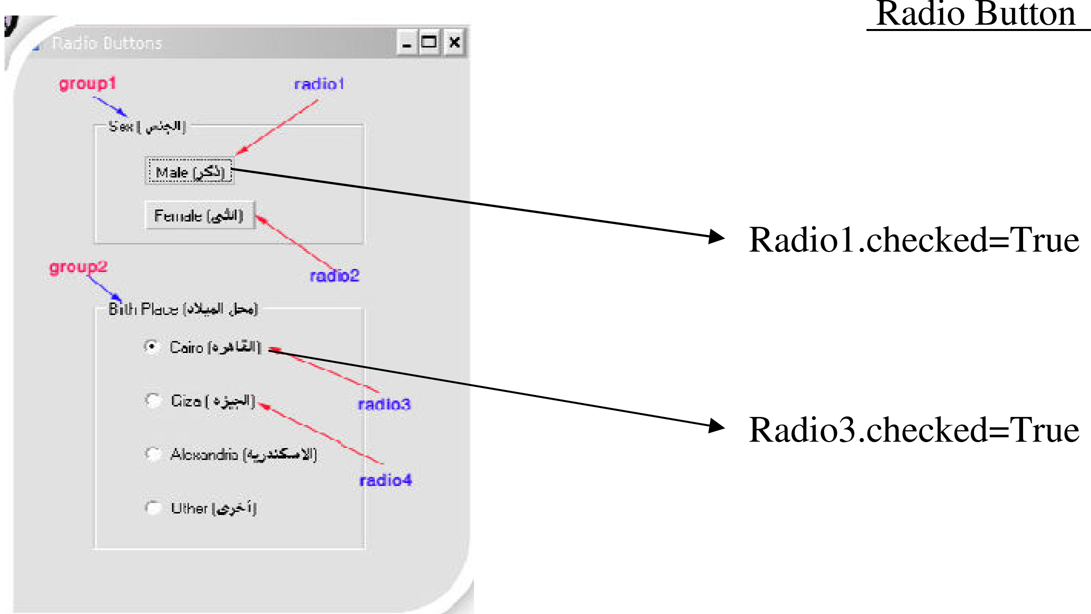
تتيح للمستخدم مجموعة من البدائل لاختيار واحدة منها فقط .

- **<u>خصائص الأداة Radio Button</u>**

| الخاصية (property) | العمل (Action) |
|---|---|
| Name | تحديد اسم الأداة وعادة ما يبدء الإسم بالأحرف (rad) |
| BackColor | تحديد لون خلفية الأداة |
| Appearance | تحديد ما إذا كان شكل الأداة هو الشكل التقليدى لزر الراديو أو أنه شكل زر عادى. |
| Checked | تحديد ما إذا كانت الأداة قد تم اختيارها أم لا (False أو True ) |
| Enabled | تحديد ما إذا كان المستخدم يستطيع التعامل مع الأداة أم لا |
| ForeColor | تحديد لون الكتابة على الأداة |
| Height | تحديد ارتفاع الأداة |
| Image | تحديد الصورة التى ستظهر على الأداة |
| Text | تحديد الكتابة التى ستظهر على الأداة |
| Visible | تحديد ما إذا كانت الأداة مرئية أم غير مرئية |
| Width | تحديد أتساع الأداة |

<u>مثال علي أشكال Radio Button</u>



- **الاداة CheckBox**

تتيح للمستخدم مجموعة من البدائل لاختيار أي عدد منها

- **خصائص الاداة CheckBox**

| الخاصية | العمل |
|---|---|
| Name | استرجاع أو تحديد اسم الأداة وعادة ما يبدء الإسم بالأحرف (chk) |
| BackColor | تحديد لون خلفية الأداة |
| Appearance | تحديد ما إذا كان شكل الأداة هو الشكل التقليدى مربع صغير أو أنه شكل زر عادى |
| Checked | تحديد ما إذا كانت الأداة قد تم اختيارها أم لا (False أو True ) |
| Enabled | تحديد ما إذا كان المستخدم يستطيع التعامل مع الأداة أم لا |
| ForeColor | تحديد لون الكتابة على الأداة |
| Height | تحديد ارتفاع الأداة |
| Image | تحديد الصورة التى ستظهر على الأداة |
| Text | تحديد الكتابة التى ستظهر على الأداة |
| Visible | تحديد ما إذا كانت الأداة مرئية أم غير مرئية |
| Width | تحديد أتساع الأداة |

**مثال**

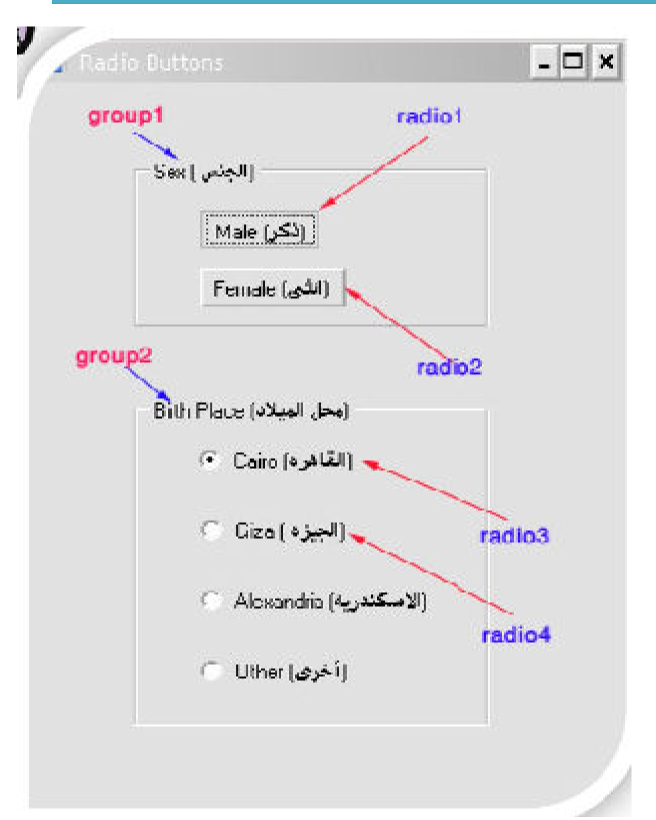
☑ Engineer
☐ Doctor
☑ Officer
☐ Lawer
☐ Business man

- **الاداة GroupBox**
  - تستخدم لتنظيم الأدوات علي النموذج .
  - تسمح للمبرمج أن يقسم عدد من الأداة Radio Button و Checked Box إلي مجموعات ويقوم المستخدم باختيار واحدة فقط من كل مجموعة .

- **خصائص الاداة GroupBox**

| الخاصية (Property) | العمل (Action) |
|---|---|
| Name | استرجاع أو تحديد اسم الأداة وعادة ما يبدء الإسم بالأحرف (grp) |
| Enabled | تحديد ما إذا كان المستخدم يستطيع التعامل مع الأداة أم لا ، أي أن الأداة مفعلة أم لا. في حالة عدم تفعيل الأداة ( Enabled = False ) فإن جميع الأدوات داخل الـ GroupBox تصبح غير مفعلة. |
| Flat Style | استرجاع أو تحديد شكل الأداة Group Box ويوجد لدينا أربعة اختيارات لهذه الأداة وهي "System , ( الافتراضية ) Flat, PopUp, Standard " |
| Visible | تحديد ما إذا كانت الأداة مرئية أم غير مرئية. في حالة كون الأداة غير مرئية (Visible = False ) فان جميع الأدوات داخل الـــGroupBox تصبح غير مرئية |

**مثال**




Group1.Text="sex( الجنس )"
Group2.Text="Birth Place( محل الميلاد )"

# أسئلة الكتاب للفصل الثاني

**السؤال الأول : اختر الإجابة الصحيحة مما يلي :**

**١- الأمر IF...Then...Else هو أمر اختيار ..........................**

أ) فردى

**ب) مزدوج**

ج) ثلاثي

د) لاشيء مما سبق

**٢- يطلق على الأمر IF...Then أمر .................... لأنه يقوم بتحديد أو تجاهل عمل واحد .**

**أ) Single – Selection**

ب) Double – Selection

ج) Multiple – Selection

د) Repletion

**٣- استخدام المعامل ........... يؤدى إلى خطأ العملية إذا كان ما على يساره اقل مما على يمينه**

أ) =

**ب) ≥**

ج) ≤

د) لاشيء مما سبق

**٤- جملة الأمر IF...Then...Else تنتهي بكلمات ..................**

**أ) End IF**

ب) End Else

ج) EndIF

د) EndElse

**٥- الجملة التالية يتم تنفيذها فقط في حالة أن يكون الشرط يساوى :**

```
IF ( Condition )
        MsgBox("n is divisible by d")
End IF
```

أ) ( n mod d ) < >0

ب) ( n \ d ) = 0

**ج) ( n mod d ) = 0**

د) ( n \ d ) < > 0

**٦- الخاصية ................. تقوم بإعطاء قيمة لملصق الــ CheckBox**

أ) Label

**ب) Text**

ج) Checked

د) Name

**٧- عدد Check Boxes الممكن اختيارها في المرة الواحدة هو ..............**

أ) واحد

ب) اثنان

ج) ثلاثة

**د) أي عدد ممكن تحديده**

**٨- الشرط ( a And Also b ) يكون صحيحاً إذا كان :**

أ) a خطأ و b أيضاً خطأ

ب) a خطأ و b صحيحاً

ج) a صحيحاً و b خطأ

**د) a صحيحاً و b أيضاً صحيحاً**

**٩- الشرط ( a Or b ) يكون خطأ إذا كان :**

أ) a خطأ أو b خطأ

ب) a خطأ و b صحيحاً

ج) a صحيحاً و b خطأ

**د) جميع ما سبق**

**١٠- الشرط ( m Or n ) يكون صحيحاً إذا كان :**

أ) m صحيحاً و n أيضاً صحيحاً

ب) m خطأ و n صحيحاً

ج) m صحيحاً و n خطأ

**د) الجملتان (ب) و (ج) صحيحتان**

**السؤال الثاني : بفرض أن المستخدم أدخل القيمة ( 73.5 ) في مربع للكتابة اسمه txtScore حدد تأثير هــذه العملية على الكود الآتي :**

```
Dim Score As Decimal
Score = Decimal . Parse ( txtScore . Text )
IF Score >= 90 Then
lblDisplay . Text = "Excellent"
Else IF Score >= 75 Then
lblDisplay . Text = "Very Good"
Else IF Score >= 60 Then
lblDisplay . Text = " Good"
Else IF Score >= 50 Then
lblDisplay . Text = " Pass"
Else
lblDisplay . Text = " Fail"
End IF
```

**الإجابة :**

**يقوم هذه الكود بطباعة كلمة Good داخل المبين lblDisplay**

**السؤال الثالث : بفرض أن lblAmPm هي أداة ملصق ( Label Control ) فما الخطأ الموجود في الكود التالي :**

```
Dim Hour As Integer
Hour = 13
If Hour ≤ 11
lblAmBm.Text = "Am"
Else
lblAmBm.Text = "Pm"
Elseif Hour > 23 Then
lblAmBm.Text = "hour can not be more than 23"
End IF
```

**الإجابة**

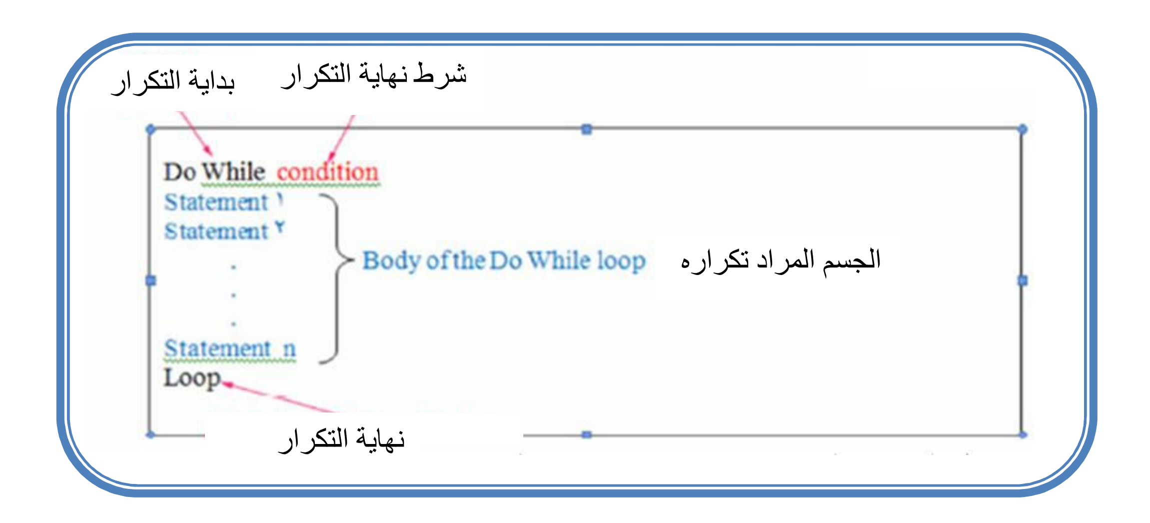
```
Dim Hour As Integer
Hour = 13
If Hour > 23 Then
lblAmBm.Text = "hour can not be more than 23"
Else If Hour <= 11 Then
lblAmBm.Text = "Am"
Else
lblAmBm.Text = "Pm"
End IF
```

# الفصل الثالث
# أوامر التكرار

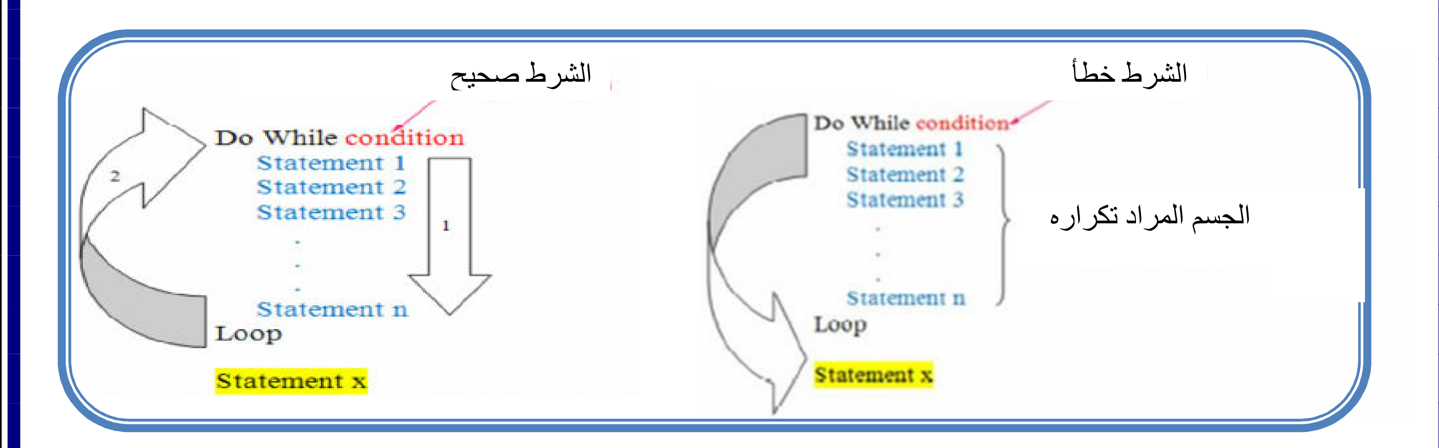
- في هذا الفصل سيتم تقديم **أوامر التكرار** وهي أوامر تتحكم في إعادة تنفيذ عمل معين أو مجموعة من الأوامر حتى يتحقق شرط معين .

- **الأمر Do While…………Loop** ( نفذ بينما …….. )

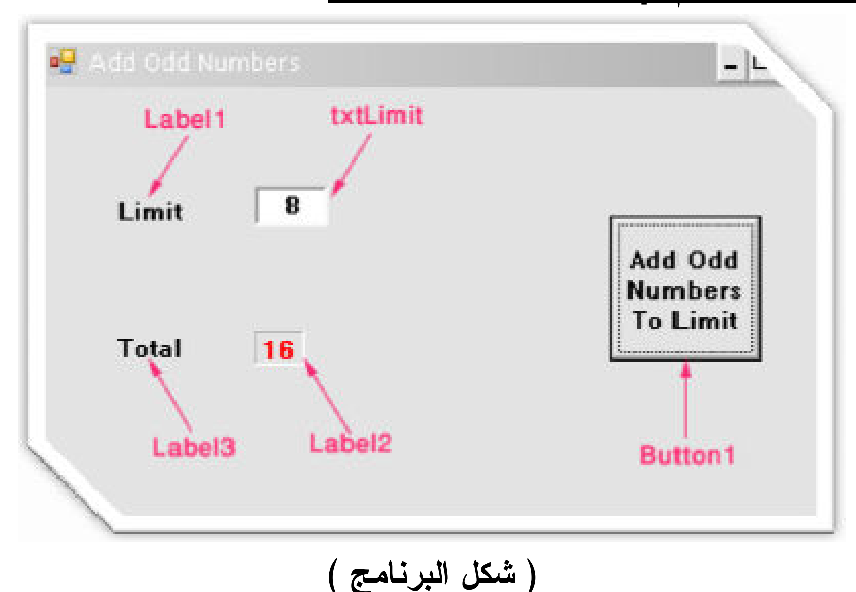
يتم إتباع القاعدة التالية :




- يستخدم هذا الأمر لتكرار أمر معين أو مجموعة من الأوامر حتى يتحقق شرط معين.
- في هذا الأمر لا نعرف عدد المرات التي سيتم تكرارها لأن عدد مرات التكرار هنا يعتمد على شرط معين
- يتم تنفيذ الأمر Do While …. Loop طالما الشرط متحقق وعندما لا يتحقق الشرط يتم الخروج من الدوارة وتنفيذ ما بعدها مباشرة . وفى حالة عدم تحقق الشرط لا يتم تنفيذ أي أوامر داخل الدوارة ويتم الخروج منها وتنفيذ ما بعدها .
- في كل مرة يصل فيها البرنامج إلى Do While يقوم باختبار الشرط فإذا كان الشرط خطأ False يتم تنفيذ أول أمر بعد كلمة Loop مباشرة ، أما إذا كان الشرط صحيح True يتم تنفيذ مجموعة الجمل التي تلي الشرط وعندما يصل البرنامج إلى كلمة Loop يعود مرة أخرى إلى Do While لاختبار الشرط مرة أخرى وهكذا .




**مثال : أكتب كود برنامج يستخدم لجمع الأعداد الفردية من 1 إلى Limit محدد يتم إدخاله إلى البرنامج أثناء تشغيله داخل TextBox باستخدام Do While ... Loop :**




( شكل البرنامج )

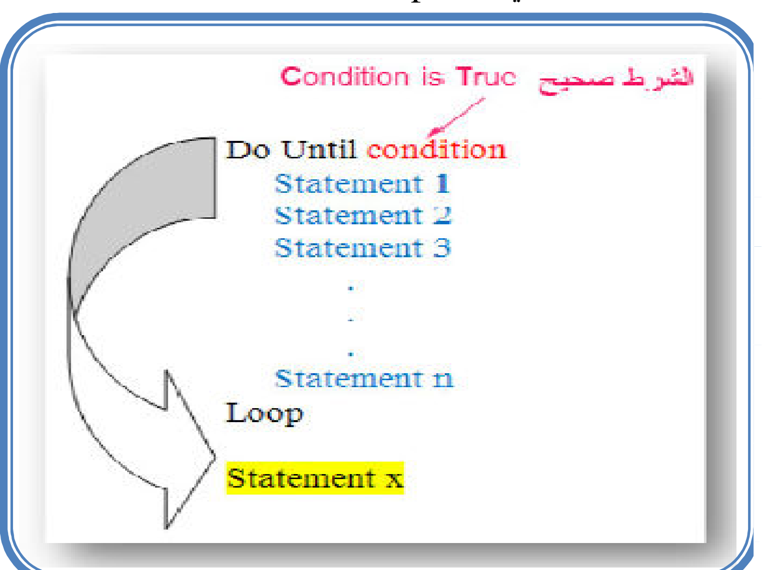
```
1    Dim X , Sum , Limit
2    Limit = Integer. Parse (txtLimit.Text)
3    X = 1
4    Do While X < = Limit
5    Sum = Sum + X
6    X = X + 2
7    Loop
8    Print Sum
```

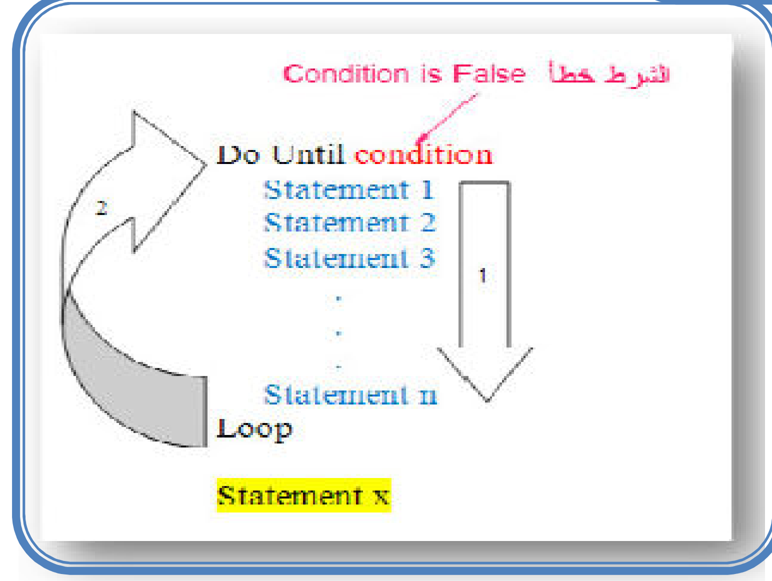
- تعريف المتغيرات (السطر 1)
- تحديد المدخلات (السطر 2)
- الجسم المراد تكراره (الأسطر 3 – 6)

( كتابة الكود )

- **الأمر Do Until ........ Loop** ( نفذ حتى ........ )

على العكس من Do While ........ Loop فان Do Until ........ Loop يختبر الشرط فإذا كان خاطئاً False يبدأ التكرار ويستمر حتى يصبح الشرط صحيحاً True وعندها يتوقف التكرار،
والشرط الموجود في البناء Do Until ........ Loop يسمى شرط إنهاء التكرار .



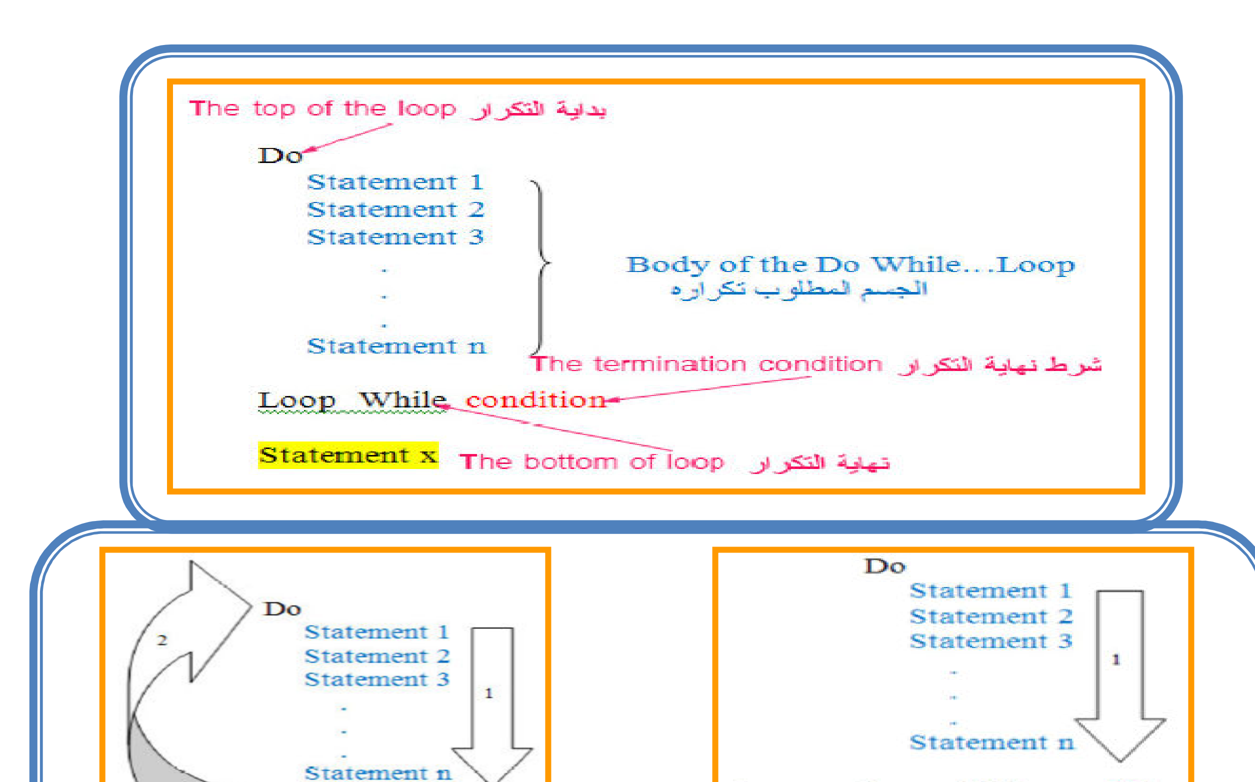

**مثال : نفس المثال السابق**

**أكتب كود برنامج يستخدم لجمع الأعداد الفردية من 1 إلى Limit محدد يتم إدخاله إلى البرنامج أثناء تشغيله داخل TextBox باستخدام Do Until ... Loop :**

```
Dim X , Sum , Limit
Limit = Integer. Parse (txtLimit.Text)
X = 1
Do Until X > Limit
Sum = Sum + X
X = X + 2
Loop
Print Sum
```

## • الأمر Do .......... Loop While

يعمل الأمر Loop While .......... Do مثل الأمر Loop ........ Do While مع اختلاف بسيط وهو أن الشرط يتم تقييمه بعد نهاية كل تكرار وليس قبل بدايته وهذا يؤدي إلي أن الأوامر التي ستتكرر سيتم تنفيذها على الأقل مرة واحدة بعكس Loop ........ Do While حيث أن الأوامر قد لا يتم تنفيذها علي الإطلاق .




الشرط صحيح

الشرط خطأ

**مثال : نفس المثال السابق**

**أكتب كود برنامج يستخدم لجمع الأعداد الفردية من 1 إلى Limit محدد يتم إدخاله إلى البرنامج أثناء تشغيله داخل TextBox باستخدام Do ... Loop While :**

```
Dim X , Sum , Limit
Limit = Integer. Parse (txtLimit.Text)
X = 1
Do
 Sum = Sum + X
X = X + 2
Loop While X <= Limit
Print Sum
```

## • الأمر Do .......... Loop Until

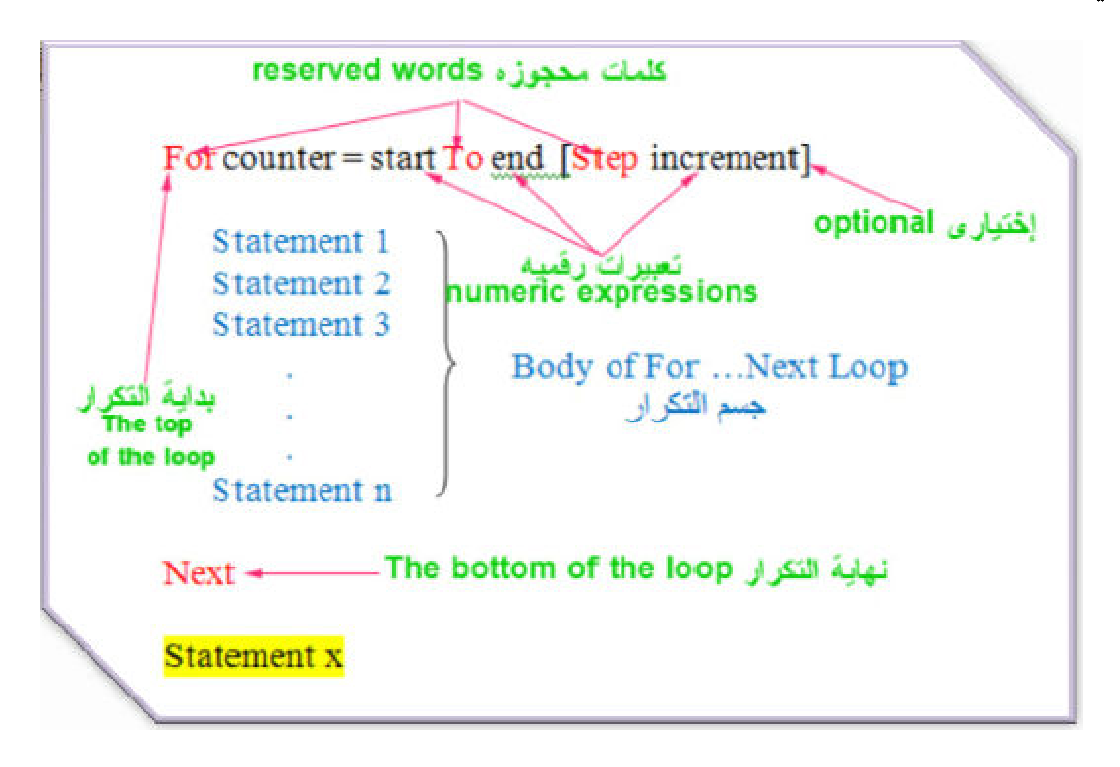
نفس أوجه الشبة وأوجه الاختلاف بين Loop ........ Do While و Loop ........ Do Until موجود أيضا بين Loop While .......... Do و Loop Until .......... Do .

**<u>مثال : نفس المثال السابق</u> أكتب كود برنامج يستخدم لجمع الأعداد الفردية من 1 إلى Limit محدد يتم إدخالـــه إلى البرنامج أثناء تشغيله داخل TextBox باستخدام Do ... Loop Until :**

```
Dim X , Sum , Limit
Limit = Integer. Parse (txtLimit.Text)
X = 1
Do
 Sum = Sum + X
X = X + 2
Loop Until X > Limit
Print Sum
```

## • الأمر For .......... Next

على خلاف أوامر التكرار السابقة والتي لا نعرف فيها عدد مرات التكرار فان الأمر For .......... Next يستعمل في حالة معرفة عدد مرات التكرار مسبقاً .




<u>حيث</u>

- تحدد الكلمة **For** بداية التكرار
- وتحدد الكلمة **Next** نهاية التكرار
- والعداد **Counter** هو متغير رقمي يستعمله المبرمج لحساب عدد مرات التكرار مع ملاحظة أن الكلمات ( **Start End – Increment** ) هي قيم رقمية يتولى المبرمج تحديدها .
- الكلمة **Step** استعمالها اختياري وفى حالة عدم استخدامها تزيد قيمة العداد بمقدار **1** بعد كل تكرار .

# أمثلة متنوعة

**مثال ١ :**

أكتب كود برنامج يقوم بتغيير قيمة العداد **Counter** من **5** إلى **50** بزيادة **2** في كل مرة باستخدام أمر **For…Next** .

```
For Counter = 5 To 50 Step 2
```

**مثال ٢ :**

أكتب كود برنامج يقوم بتغيير قيمة العداد **i** من **1** إلى **100** بزيادة **1** في كل مرة باستخدام أمر **For…Next** .

```
For i = 1 To 100 Step 1
Or
For i = 1 To 100
```

**مثال ٣ :**

أكتب كود برنامج يقوم بتغيير قيمة العداد **i** من **10** إلى **2** بإضافة **-2** في كل مرة باستخدام أمر **For…Next** .

```
For i = 10 To 2 Step -2
```

**مثال ٤ :**

أكتب كود برنامج يقوم بتغيير قيمة العداد **i** كالتالي ( **19 , 17 ,15 , 13 , 11** ) باستخدام أمر **For…Next**.

```
For i = 11 To 19 Step 2
```

**مثال ٥ :**

أكتب كود برنامج يقوم بتغيير قيمة العداد **i** كالتالي ( **21 , 28 ,35 , 42 , 49** ) باستخدام أمر **For…Next**.

```
For i = 49 To 21 Step -7
```

**مثال ٦ :**

اكتب كود برنامج لحساب مجموعة الأعداد الزوجية من **2** إلى **100** وطباعة ناتج الجمع باستخدام **DoWhile….Loop** .

**الحل**

سنقوم بتعريف متغيرين من النوع **Integer** هما **N** و **Sum** الأول كعداد والثاني كمتغير للمجموع والقيمة الافتراضية أو الابتدائية لهم هي صف ثم نكتب البرنامج كالتالي :

1. Dim N, Sum as Integer
2. Do While ( N < 100 )
3. N = N +2
4. Sum = Sum + N
5. Loop
6. Print Sum

**مثال ٧**

أكتب كود برنامج لطباعة الأعداد الصحيحة من **10** إلى **15** باستخدام **For ….Next**

**الحل**

1. Dim N as Integer
2. For N = 10 to 15
3. Print N
4. Next

مثال٨

أكتب كود برنامج لطباعة الأعداد الصحيحة الزوجية من 10 إلى 20 باستخدام For ....Next

الحل

1. Dim N as Integer
2. For N = 10 to 20 step 2
3. Print N
4. Next

مثال٩

أكتب كود برنامج لطباعة الأعداد الصحيحة الزوجية من -10 إلى -20 باستخدام For ....Next

الحل

1. Dim N as Integer
2. For N = -10 to -20 step -2
3. Print N
4. Next

مثال١٠

أكتب كود برنامج لطباعة الأعداد الصحيحة الفردية من 1 إلى 99 باستخدام For ....Next

الحل

1. Dim N as Integer
2. For N = 1 to 99 step 2
3. Print N
4. Next

# الدالة Format ( )

تستخدم لتحديد شكل المخرجات حيث يمكن استخدام العلامة العشرية وفاصل الآلاف وشاغلات أماكن الأرقام مثل ( 0 و # )

- عند استخدام 0 يستبدل به رقم أو صفر في المخرجات

مثال : أكتب ناتج الأمر التالي Format ( 5.5 , "00.00" )

الحل : ناتج الدالة السابقة هو : 05.50

- عند استخدام # يستبدل به رقم أو لاشيء في المخرجات

مثال : أكتب ناتج الأمر التالي Format ( 5.5 , "##.##" )

الحل : ناتج الدالة السابقة هو : 5.5

| Format function | Result for x = 3.4 | Result for x = 3.456 |
|---|---|---|
| Format(x) | 3.4 | 3.456 |
| Format(x, "00.00" ) | 03.40 | 03.46 |
| Format(x, "##.##") | 3.4 | 3.46 |
| Format(x, "##.00" ) | 3.40 | 3.46 |
| Format(x, "00.##" ) | 03.4 | 03.46 |

تدريب

| الدالة | النتيجة |
|---|---|
| **Format (3.4)** | |
| **Format (6.871)** | |
| **Format (2.2, "00.00")** | |
| **Format (2.111, "00.00")** | |
| **Format (8.4, "##.00")** | |
| **Format (66.33, "###.##")** | |
| **Format (3.4, "00.#")** | |
| **Format (13.40, "0#0.0#0")** | |

## الأداة ListBox

- أحد الأدوات التي ترتبط بالتكرار وتمكن من إنشاء قائمة من العناصر للاختيار منها .
- تظهر هذه الأداة على شكل مستطيل يعرض بداخله صفوفاً من الكلام يمثل كل صف عنصراً يمكن لمستخدم التطبيق الاختيار منها .
- عند وجود عناصر بالقائمة أكثر مما يظهر في مساحة الأداة يظهر شريط تمرير يمكن عن طريقه من الوصول لباقي عناصر الأداة . والشكل التالي يوضح صورة لها




## خصائص الأداة ListBOX

| الخاصية | وظيفتها |
|---|---|
| Name | تستخدم لتحديد اسم الأداة داخل نافذة الكود ويفضل أن يبدأ اسم هذه الأداة بالأحرف lst |
| Items | تستخدم هذه الخاصية لإضافة العناصر داخل الأداة ListBox وهذه الخاصية في حد ذاتها كائن Object له خصائص ووظائف |
| SelectedItems | تستخدم لتحدد العنصر الواقع عليه الاختيار في الأداة ListBox والمقصود معرفة العنصر الذي قام المستخدم باختياره |
| Sorted | هذه الخاصية تستخدم لتحديد ما إذا كان سيتم ترتيب القيم الموجودة بالقائمة العرض أبجديا أم لا ، بصرف النظر عن ترتيب إدخالها ولها قيمتان :<br>• False : (القيمة الافتراضية) وتظهر القيم داخل قائمة العرض بنفس ترتيب إضافتها إلى قائمة العرض .<br>• True : تظهر القيم داخل قائمة العرض مرتبة أبجدياً بصرف النظر عن ترتيب إدخالها . |

## طرق ووظائف Method الخاصية Items للأداة ListBox

- الطريقة أو الوظيفة أو الوسيلة Method هي وصف لسلوك معين مصاحب للكائن بمعنى أنها أفعال محددة مسبقاً لكل كائن .
- طريقة كتابة وسيلة كائن هى : ObjectName . Method ( )

| الطريقة Method | العمل |
|---|---|
| Add (text) | إضافة ما بين القوسين من نص في آخر عناصر القائمة بمعنى إضافة عنصر جديد للقائمة |
| Count | معرفة عدد العناصر في القائمة |
| Insert ( Index , text ) | إضافة العنصر text في المكان المحدد بـ Index فالعناصر الموجودة عند المكان الذي له الرقم Index وحتى آخر القائمة تتحرك لأسفل خطوة واحدة لإخلاء مكان للعنصر الجديد |
| Remove (text) | البحث فى عناصر القائمة عن العنصر الذى يساوى TEXT تماما وفى حالة وجوده تتم إزالته وتتحرك كل العناصر تحته إلى أعلى خطوة واحدة لتملأ الفراغ وإذا لم يوجد عنصر يساوى TEXT فلا يحدث شيء |
| RemoveAt (index) | إزالة العنصر المحدد مكانه بالرقم Index مع تحريك كل العناصر الموجودة تحته إلى أعلى خطوة واحدة لتملأ الفراغ |
| Clear ( ) | إفراغ الأداة ListBox من كل العناصر بمعنى إزالة كل العناصر الموجودة بها |

### ملحوظات هامة جداً

- المعامل Index يأخذ قيمة رقمية تتراوح من صفر إلى ( Count – 1 ) حيث أن Count يمثل عدد العناصر فى الأداة ListBox .
- وبالتالي فان المعامل Index يبدأ ترقيمه بصفر فالعنصر الأول داخل ListBox يأخذ الرقم 0 كترتيب والثاني 1 والثالث 2 وهكذا ........

### الثابت ControlCharts.Tab

- الثابت ControlCharts.Tab هو أحد ثوابت VB.Net ويستخدم لترك مسافة قدرها Tab أي 6 مسافات بالمسطرة بين القيم وبعضها .

# حل أسئلة الكتاب الفصل الثالث

**السؤال الأول : اختر الإجابة الصحيحة مما يلي :**

1- الأوامر المتكررة في Do While...Loop :
أ) لا يتم تنفيذها بالمرة
ب) يتم تنفيذها مرة على الأقل
ج) يتم تنفيذها إذا كان الشرط صحيحاً "True"
د) يتم تنفيذها إذا كان الشرط خاطئاً "False"

2- الأوامر المتكررة في Do Until...Loop يتم تنفيذها إذا استمر الشرط الآتي زائفاً False :
أ) شرط الـ Do-Loop .
ب) شرط استمرار التكرار .
ج) شرط الـ Until-Loop .
د) شرط نهاية التكرار .

3- نوعية الأمر الذى يتم تنفيذه حتى يصبح شرط التكرار خطأً هى :
أ) Do Until ... Loop
ب) Do
ج) Do While
د) Do While ... Loop

4- يطلق على المتغير الذى يتحكم فى عدد مرات تنفيذ الأوامر اسم
أ) repeater
ب) Loop
ج) Counter
د) لا شيء مما سبق

5- لإلغاء كل القيم من الأداة ListBox يتم استخدام الطريقة Method التالية :
أ) Delete
ب) Clear
ج) Remove
د) Destroy

6- الطريقة Method التي تقوم بإضافة عنصر جديد الأداة ListBox هى :
أ) Add
ب) Insert
ج) Include
د) Append

**السؤال الثاني : قم بدراسة الأوامر التالية بفرض أن المتغيرات قد تم الإعلان عنها**

```
1-    Sum = 0
2-    Count = 0
3-    Do While ( X < 10 )
4-    Sum = Sum + X
5-    Count = Count + 1
6-    Loop
7-    Average = Sum / Count
```

- ما الخطأ في هذه الأوامر
- اكتب الخطوات اللازمة لتصويب هذه الأوامر

الحل

الخطأ في هذه الأوامر أنه لم تتم زيادة المتغير X داخل Do Loop فتوضع قبل معادلة Sum والحل الصحيح كما يلي :

```
1-    Sum = 0
2-    Count = 0
2-    X = 0
4-    Do While ( X < 10 )
5-    X = X + 1
6-    Sum = Sum + X
7-    Count = Count + 1
8-    Loop
9-    Average = Sum / Count
```

**السؤال الثالث : قم بالبحث عن الخطأ (الأخطاء) في البرنامج التالي بافتراض أن التكرار يقوم بجمع الأرقام من 1 إلى 10 .**

```
1    Dim X As Integer = 1
2    Dim Sum As Integer = 0
3    Do Until X < = 10
4    Sum = Sum + X
5    X = X + 1
6    Loop
```

الحل :

الخطأ في البرنامج السابق هو شرط التكرار حيث أن Do Until تعمل عندما يكون الشرط خطأ ولكي يكون البرنامج صحيح يكون بالشكل :

```
Dim X As Integer = 1
Dim Sum As Integer = 0
Do Until X > 10
Sum = Sum + X
X = X + 1
Loop
```

**<u>السؤال الرابع</u>** <u>: **افترض مجموعة الأوامر التالية**</u> :

```
For I = 1 To 10 Step X
MsgBox( i )
Next
```

<u>اشرح كيفية تنفيذ الأوامر فى كل حالة من الحالات الآتية :</u>

- إذا كان المتغير **X** أكبر من صفر
- إذا كان المتغير **X** أصغر من صفر
- إذا كان المتغير **X** يساوى صفر

**<u>الحل</u>**

<u>1- إذا كان المتغير **X** أكبر من صفر وليكن قيمته هي 1 أي **X = 1** ليصبح البرنامج كالتالى</u>

```
For I = 1 To 10 Step 1
MsgBox( i )
Next
```

يطبع البرنامج الأعداد الطبيعية من 1 إلى 10 .

<u>2- إذا كان المتغير **X** أصغر من صفر وليكن **X = -1** ليصبح شكل البرنامج كالتالى :</u>

```
For I = 1 To 10 Step -1
MsgBox( i )
Next
```

هنا Step خطأ لأن البرنامج لن ينتهي حيث أن الزيادة في طريق عكسي بالسالب ولن نصل إلى النهاية

<u>3- إذا كان المتغير **X** يساوى صفر سيصبح شكل البرنامج</u>

```
For I = 1 To 10 Step 0
MsgBox( i )
Next
```

بالطبع لا يصلح أن تكون الزيادة بصفر وهذا خطأ داخل البرنامج والبرنامج لن يعمل .

**السؤال الخامس : قم بتتبع مجموعة الأوامر الآتية موضحاً قيمة كل متغير فى كل مرة يتغير فيها**

1- X = 4.5
2- For i = 0 To 7 Step 3
3- X = X * 2
4- Next

**الحل**

| الأمر | i | X |
|---|---|---|
| X = 4.5 | غير معروفة حتى الآن | 4.5 |
| For i = 0 To 7 Step 3 | 0 | 9 |
| | 3 | 18 |
| | 6 | 36 |

**السؤال السادس : قم بتحويل الأمر For … Next التالي إلى الأمر Do While …. Loop مع تغيير ما يلزم**

1- For i = 1 To 10
2- Msg Box ( i * i )
3- Next

**الحل**

1- i = 1
2- Do While ( i <=10)
3- Msg Box ( i * i )
4- i = i + 1
5- Loop

## السؤال السابع : صف المخرجات من تنفيذ مجموعة الأوامر التالية

```
For i = 0 To 7
if ( i % 2 = 0 ) Then
MsgBox( i + 1 )
Else If ( i % 3 = 0 ) Then
MsgBox( i * i )
Else If ( i % 5 = 0 ) Then
MsgBox( 2 * i - 1 )
Else
MsgBox( i )
        End If
        Next
```

## الحل

ملحوظة هامة : العلامة % تساوى mod وهى بمعنى باقى القسمة

| الأمر | i | الأمر الذي سينفذ | خرج البرنامج |
|---|---|---|---|
| For i = 0 To 7 | 0 | MsgBox( i ) | 0 |
| | 1 | MsgBox( i ) | 1 |
| | 2 | MsgBox( i + 1 ) | 3 |
| | 3 | MsgBox( i * i ) | 9 |
| | 4 | MsgBox( i + 1 ) | 5 |
| | 5 | MsgBox( 2 * i - 1 ) | 9 |
| | 6 | MsgBox( i + 1 ) | 7 |
| | | MsgBox( i * i ) | 36 |
| | 7 | MsgBox( i ) | 7 |

# الفصل الرابع

## الميقاتى : Timers

**الميقاتى : Timer يطلق عليه كائن الحدث الواحد**

- الكائن Timer هو ساعة توقيت غير مرئية تتيح التعامل مع ساعة النظام System Clock من داخل البرنامج التي تقوم بتنفيذها .
- هذه الكائن أو الأداة تعمل في الخلفية فلا يشعر بها أو يراها المستخدم .
- لأن هذه الأداة لا تظهر إلا في التصميم فقط فهي ليست في حاجة للخصائص التي تحدد المظهر العام مثل Height و Width . ولها سبعة خصائص فقط .
- الكائن Timer يعمل على استثارة حدث واحد فقط هو الحدث Tick ويتم استثارة هذا الحدث بعد مرور فترة زمنية محددة بواسطة الخاصية Interval وهى أهم خصائص الأداة Timer .
- قيمة الخاصية Interval (المدة الزمنية) تتراوح من 1 إلى 65535 وتقاس بوحدة الميللى ثانية .
- الميللى ثانية = واحد من الألف من الثانية والجدول التالي يوضح ذلك :

| قيمة الخاصية Interval | ما يعادلها بالثانية |
|---|---|
| 1000 | 1 ثانية |
| 2000 | 2 ثانية |
| 500 | 0.5 ثانية |

- ينشط الكائن Timer على فترات زمنية متساوية تسمى هذه الفترات Intervals ويقع الحدث Tick كنتيجة لهذا النشاط .
- لينشط الكائن Timer حسب الفترة الزمنية المحددة بالخاصية Interval نقوم بضبط الخاصية Enable عند القيمة True . حيث أن الخاصية Enable تسمى خاصية التمكين .
- لمنع وقوع الحدث Tick أو إيقاف عمل الأداة أو الكائن Timer عن العمل نعطى الخاصية Enable القيمة False .
- القيمة الافتراضية للخاصية Enable هي False وذلك للأداة Timer .

## التكوين DateTime

- عند الإعلان عن متغير من النوع Date يصبح بالإمكان استعمال التكوين DateTime وهـذا التكـوين يتمتـع بخصائص وطرق (وسائل) .
- والجدول التالي يوضح بعض خصائص التكوين DateTime بفرض أنه تم الإعلان عن متغير من النوع Date اسمه X كما يلي :

**Dim X As Date**

| الخاصية | الغرض | مثال | النتيجة |
|---|---|---|---|
| Now | استرجاع التاريخ والوقت الحالي | X = Now | 8/2/2009<br>11:49:50 AM |
| Date | إدراج التاريخ | X.Date | 8/2/2009 |
| Day | إدراج رقم اليوم خلال الشهر<br>1 - 31 | X.Day | 8 |
| DayOfYear | إدراج رقم اليوم خلال السنة<br>1 - 366 | X.DayOfYear | 8 |
| Hour | إدراج الساعة<br>0 - 23 | X.Hour | 11 |
| Minute | إدراج الدقيقة<br>0 - 59 | X.Minute | 49 |
| Second | إدراج الثانية<br>0 - 59 | X.Second | 50 |
| Month | إدراج اسم الشهر ويستبدل الاسم برقم حيث أن 1 مثلا تعنى الشهر الأول وهو يناير | X.Month | 2 |

### بعض الطرق لتعديل الوقت والتاريخ

| الطريقة | الغرض | مثال | النتيجة |
|---|---|---|---|
| AddDays | إنشاء تاريخ جديد متقدم أو متأخر عدد من الأيام | X.AddDayes(1)<br>X.AddDayes(-2) | تقديم يوم واحد<br>تأخير يومان |
| AddHours | إنشاء تاريخ جديد متقدم أو متأخر عدد من الساعات | X.AddHours(5)<br>X.AddHours(-3) | تقديم 5 ساعات<br>تأخير 3 ساعات |
| AddMinutes | إنشاء تاريخ جديد متقدم أو متأخر عدد من الدقائق | X.AddMinutes(30)<br>X.AddMinutes(-15) | تقديم 30 دقيقة<br>تأخير 15 دقيقة |

### ملحوظات هامة جداً

- تستخدم الكلمة Static للإعلان عن متغير خاص ، حيث أن المتغير سيحتفظ بقيمته طول فترة التشغيل البرنـامج ولن يتم إعطاؤه قيمة ابتدائية عند كل مرة يتم فيها استدعاء البرنامج الفرعي .
- بالطبع كلمة Static هي من الكلمات المحجوزة التي لا يجوز استخدامها في غرض غير الغرض المخصص لها .

# حل أسئلة الكتاب الفصل الرابع

**السؤال الأول : اختر الإجابة الصحيحة**

1- الخاصية Interval للمؤقت Timer تحدد معدل وقوع حدث التكة Tick بالوحدة الزمنية

أ) ثانية ب) مللي ثانية ج) دقيقة د) ساعة

2- التكوين ................. يستدعي التاريخ والوقت الخاصين بالنظام

أ) CurrentTime ب) Time ج) Now د) DateTime

3- تستطيع ................. مع المتغير من النوع Date .

أ) جمع أيام ب) جمع ساعات ج) طرح أيام د) جميع ما سبق

4- لكي تطرح يوماً من المتغير X من النوع Date عليك أن تقوم بتخزين القيمة الناتجة من ............... في X .

أ) X.AddHour(-24) ب) X.SubtaractDays(1)
ج) X.AddDays(-1) د) جميع ما سبق

**السؤال الثاني**

1- ما السبب في وقوع الحدث Tick ؟

- ينشط الكائن Timer على فترات زمنية متساوية تسمى هذه الفترات Intervals ويقع الحدث Tick كنتيجة لهذا النشاط .

2- إذا كان لديك Interval قدرها 1500 فما هو مقدارها بالثواني ؟

- مقدارها 1.5 ثانية

3- ما هي وظيفة الكائن Timer ؟ وما هو الحدث المرتبط به ؟

- الكائن Timer هو ساعة توقيت غير مرئية تتيح التعامل مع ساعة النظام System Clock من داخل البرنامج التي تقوم بتنفيذها .
- ينشط الكائن Timer على فترات زمنية متساوية تسمى هذه الفترات Intervals ويقع الحدث Tick كنتيجة لهذا النشاط .